**MEDIA MONITORING**

LAPORAN PENELITIAN

Diajukan untuk Memenuhi Persyaratan Akademik dalam

Menyelesaikan Pendidikan pada Program Studi

S1 Teknik Informatika Universitas Kristen Maranatha

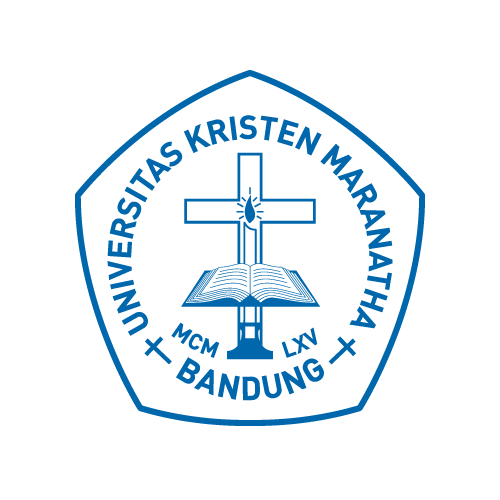
Oleh

1872033 Ray Christian Kustiawan

1972004 Yehezkiel Christian

2072042 Moses Marzuki Samosir

2072050 Puji Mei Indriana Sihombing



**PROGRAM TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS KRISTEN MARANATHA**

**BANDUNG**

**2022**

## **ABSTRAK**

Laporan ini berjudul “Media Monitoring”. Laporan ini berisi sebuah hasil laporan dari project yang diberikan oleh DailySocial.Id dalam upaya untuk melakukan media monitoring. Pada laporan ini persoalan yang dihadapi oleh penulis adalah memperoleh dan membangun dataset yang diperlukan, memproses data menjadi lebih berkualitas, dan pemilihan model yang tepat untuk memprediksi suatu topik dengan tepat. Dalam melakukan media monitoring ini penulis menggunakan TF-IDF *vectorizer* dan LDA untuk pemodelan topiknya. Lalu dari data DailySocial.Id menyediakan data internal melalui API. Selain data yang disediakan DailySocialId kami juga melakukan *web-scraping* dari hasil google search dan melakukan pengambilan data *tweet* dengan menggunakan API resmi Twitter. Penulis melakukan media monitoring ini dibantu dengan arahan dari bapak Yohanes Adhi Nugraha selaku Chief Technology Officer dari DailySocial.Id dan juga bapak Dr. Hapnes Toba, M.Sc. selaku dosen mata kuliah pencarian informasi media online.

Kata kunci : Dataset, Forecast, Monitoring, Model,

## **DAFTAR ISI**

[ABSTRAK](#_yyhd3q5hy31q) 2

[DAFTAR ISI](#_aum2eqhzoyk2) 3

[BAB I PENDAHULUAN](#_dpuoxiiyymf5) 4

[BAB II LANDASAN TEORI](#_6yqbvc5tyzlk) 6

[BAB III ANALISIS DAN DESAIN](#_mpl7a5v9itez) 11

[BAB IV HASIL PEKERJAAN](#_g9tw2br0fezr) 14

[BAB V UJI COBA SEDERHANA](#_qko72ttehub1) 19

[BAB VI KESIMPULAN DAN SARAN](#_3qnxd8px1ig) 21

[Lampiran: Jadwal Pengerjaan Proyek](#_rvw5kzg0fzko) 22

[DAFTAR PUSTAKA](#_qs1lhxtxg223) 23

## 

## **BAB I PENDAHULUAN**

1. **Latar Belakang**

Pada mata kuliah Pencarian Informasi Media Online, bekerjasama dengan DailySocialId untuk proyek yang dilaksanakan. DailySocialId adalah sebuah website penyedia berita yang berkaitan dengan teknologi dan perkembangannya. DailySocialId memerlukan *monitoring* terhadap topik yang sedang tren dan prediksinya. Selain prediksi DailySocialId juga memerlukan *monitoring* terhadap topik yang dikeluarkan kompetitor. Penyelesaian pada masalah monitoring akan membantu DailySocialId terhadap penentuan topik berita dan pembaruan berita.

1. **Rumusan Masalah**

Dari latar belakang diatas dapat diperoleh beberapa rumusan masalah yaitu antara lain :

1. Bagaimana cara memperoleh data dan membangun *dataset* yang diperlukan?
2. Bagaimana cara memproses data menjadi lebih berkualitas ?
3. Bagaimana pendekatan atau model untuk memprediksi suatu topik dengan akurat?
4. **Tujuan Pembahasan**

Tujuan dari penulisan laporan ini adalah sebagai berikut :

1. Mengetahui dan mempraktekan cara memperoleh data dan membangun *dataset* untuk kebutuhan prediksi dan *monitoring*.
2. Mengetahui cara memproses data menjadi lebih berkualitas.
3. Mencari pendekatan atau model yang digunakan untuk memprediksi suatu topik dengan akurat.
4. **Ruang Lingkup**

Ruang lingkup dari laporan ini adalah membangun *dataset* dari API resmi Twitter dan menggunakan nya untuk memprediksi tren.

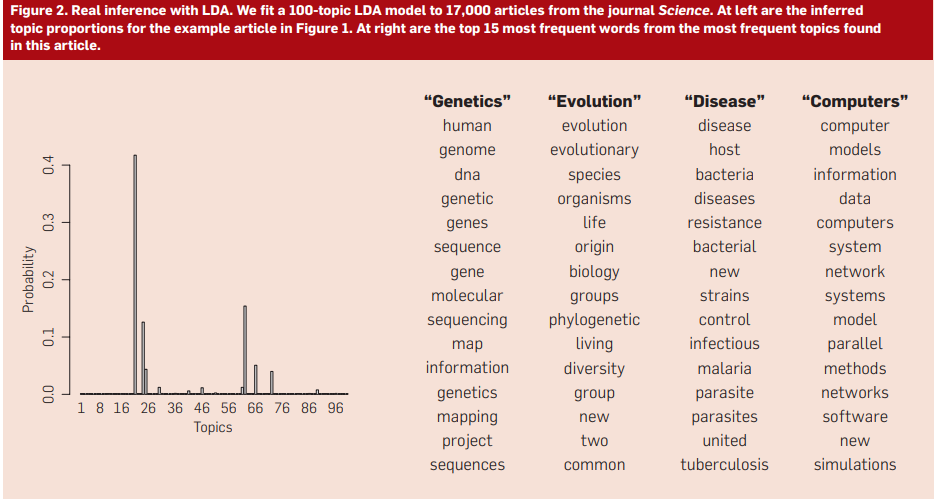
1. **Sumber Data**

Sumber data dapat berasal dari API resmi Twitter, *scrapping* halaman *web* dan data internal DailySocialId.

## **BAB II LANDASAN TEORI**

**Latent Dirichlet Allocation**

Latent Dirichlet Allocation (LDA) adalah contoh model topik dan digunakan untuk mengklasifikasikan teks dalam dokumen ke topik tertentu. membangun topik per model dokumen dan kata-kata per model topik, dimodelkan sebagai distribusi Dirichlet (Susan Li, 2018).



David M. Blei (2012:78) secara formal mendefinisikan topik sebagai distribusi melalui kosakata tetap. Dalam pemodelan probabilistik generatif, David M. Blei (2012:79) memperlakukan data sebagai yang timbul dari proses generatif yang mencakup variabel tersembunyi. Proses generatif ini mendefinisikan distribusi probabilitas bersama baik yang diamati maupun yang tersembunyi variabel acak.

**TF-IDF (Term Frequency - Inverse Document Frequency)**

TF-IDF adalah sebuah metode atau algoritma yang digunakan untuk mendapatkan bobot setiap token (kata) pada dokumen dalam korpus. Metode TF-IDF merupakan metode yang umum digunakan pada information retrieval dan TF-IDF terkenal juga sebagai metode yang mudah, efisien serta memiliki hasil yang akurat.

Rumus metode TF-IDF saat menghitung bobot setiap kata di dalam dokumen, seperti berikut :

𝑇𝐹 − 𝐼𝐷𝐹(𝑤, 𝑑) = 𝑇𝐹(𝑤, 𝑑) ∗ 𝐼𝐷𝐹(𝑤) (1)

𝐼𝐷𝐹(𝑤) = 𝑙𝑜𝑔(|𝐷|⁄𝐷𝐹 (𝑤)) (2)

Dimana:

𝑇𝐹(𝑤, 𝑑) = jumlah kata yang muncul pada dokumen 𝑑

𝐼𝐷𝐹(𝑤) = frekuensi invers dokumen kata 𝑤 dari jumlah dokumen yang ada

𝐷𝐹(𝑤) = jumlah dokumen yang memuat kata 𝑤

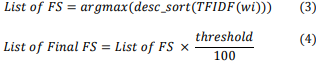
𝐷 = jumlah seluruh dokumen

Setelah bobot setiap kata dalam dokumen telah diketahui, maka selanjutnya akan dilakukan proses pengurutan dimana semakin besar bobot kata, maka semakin besar pula tingkat similaritas yang dimiliki dokumen tersebut terhadap kata kunci, dan demikian juga sebaliknya.

**Feature Selection**

Feature Selection merupakan proses pemilihan kata-kata yang nantinya akan mewakili keseluruhan kata-kata dalam proses clustering. Proses Feature Selection ini dilakukan setelah dilakukannya prose Feature Generation dan TF-IDF. Pengambilan feature selection ini dilakukan agar dimensi vektor kata menjadi lebih kecil. Kata-kata dengan bobot TF-IDF akan diurutkan secara descending dan kemudian diambil sesuai *threshold* tertentu.

Persamaan 3 dan 4 menunjukkan cara feature selection. Kata-kata inilah yang nantinya akan digunakan pada proses clustering.

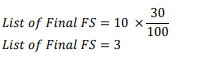


Tidak ada aturan khusus dalam penentuan *threshold,* user bisa menentukan *threshold* dalam skala 0 - 100%.

Contoh :

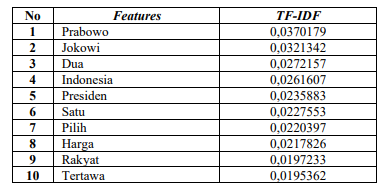
Misalnya telah didapatkan tabel feature yang diurutkan secara descending berdasarkan nilai TF-IDF seperti terlihat pada Tabel I dan user menentukan *threshold* sebesar 30%.

Maka perhitungannya adalah sebagai berikut :



Tabel I

Tabel *Features*

**

Dari perhitungan diatas, didapatkan jumlah *features* yang terpilih yaitu sebesar 3. Sehingga, mengacu pada Tabel I, maka kata-kata yang dimasukkan ke dalam vektor adalah 3 kata teratas yaitu “Prabowo”, “Jokowi”, dan “Dua”. sedangkan tujuh kata lainnya tidak dimasukkan ke dalam vektor, sehingga tidak digunakan dalam proses clustering.

**Natural language Toolkit(NLTK)**

Natural Language Processing yang juga dikenal dengan istilah Natural Language Understanding (NLU), NLP adalah subbidang ilmu komputer dan kecerdasan buatan yang berhubungan dengan interaksi antara komputer dan bahasa manusia (alami).

Terkait bidang ilmu data atau data science, kadang-kadang, NLP ini digunakan dalam penerapan algoritma machine learning pada teks dan ucapan.

NLTK menyediakan interface yang mudah digunakan ke banyak sumber daya korpora dan leksikal, mereka berisi library suite pemrosesan teks untuk klasifikasi, tokenization, stemming, tagging, parsing, dan semantic reasoning.

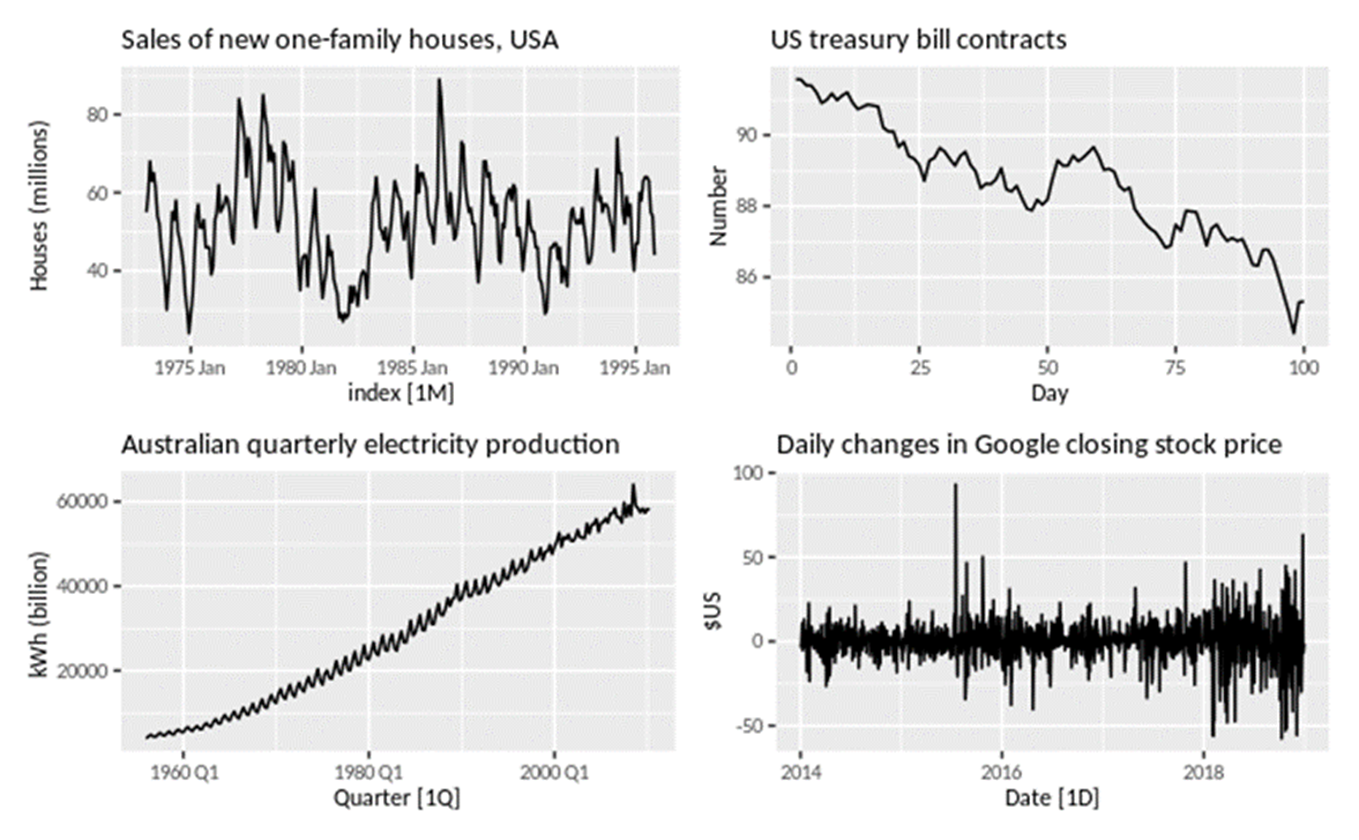
Secara umum, dapat dikatakan bahwa NLTK adalah library terbaik dalam Python,

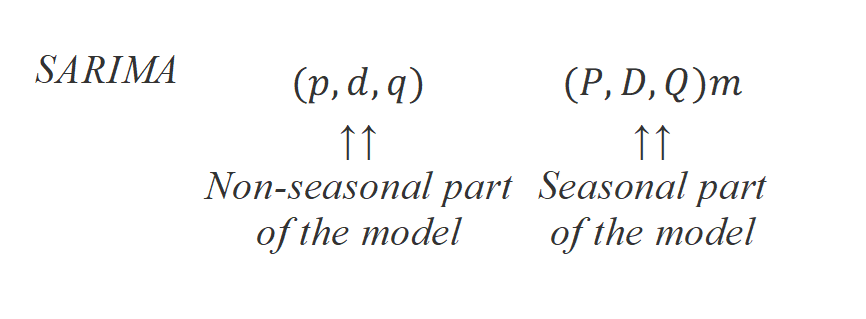
khususnya karena mereka proyek berbasis komunitas gratis, open source.

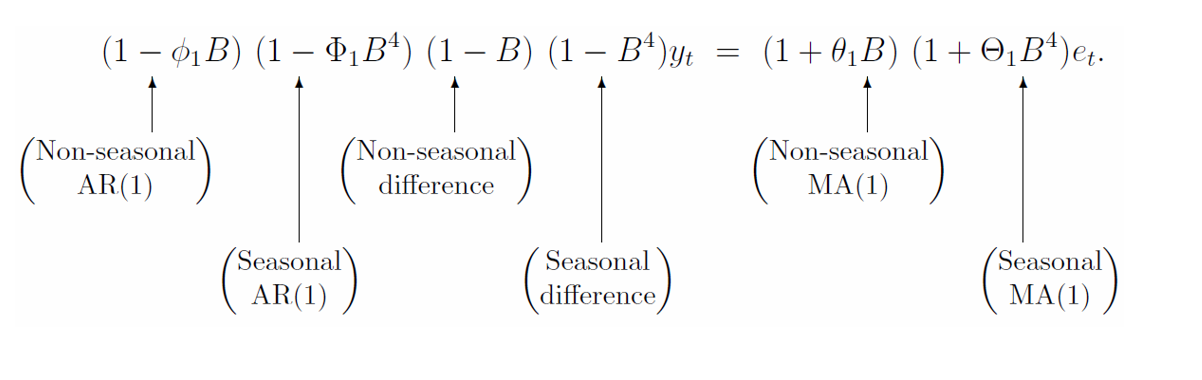
**Forecasting**

Ketika kita sudah menentukan topik yang mungkin akan menjadi tren di masa depan kita bisa melakukan forecasting untuk melihat *trend* dan *seasonality*.

*Forecasting* bisa dilakukan dengan metode ARIMA yaitu gabungan dari *Autoregressive* dan integrasi dengan *Moving Average*. Sebenarnya masih banyak cara forecasting seperti *single* hingga *triple exponential smoothing*, *Moving average* dll. Tetapi kita menggunakan ARIMA karena mencari *trend* dan *Seasonality*. Versi lebih lengkap dari ARIMA adalah SARIMA yang menambahkan parameter jangka waktu per musim pada model.







## **BAB III ANALISIS DAN DESAIN**

Untuk melakukan media monitoring diperlukan beberapa hal yaitu:

* Data
* Model
* Cara kerja/ *Workflow*

Berikut adalah uraian dari setiap elemen yang dibutuhkan:

**Data**

Untuk kebutuhan data, DailySocialId menyediakan data internal melalui API. Selain data yang disediakan DailySocialId kami juga melakukan *web-scraping* dari hasil google search dan melakukan pengambilan data *tweet* dengan menggunakan API resmi Twitter.

Penggunaan API resmi Twitter memiliki beberapa kelemahan yaitu:

* Koneksi API yang membatasi *run-time* pengambilan *tweet* sehingga tweet yang terambil sangat terbatas.
* Tweet yang sudah terambil dalam bentuk list harus di *convert* menjadi *dataframe*, ini memakan waktu yang cukup lama tergantung dari panjang *tweet*.
* API memiliki batasan koneksi tiap 15-30 menit maksudnya jika sudah mengambil 1 kali harus menunggu 15-30 menit untuk pengambilan *batch tweet* selanjutnya.

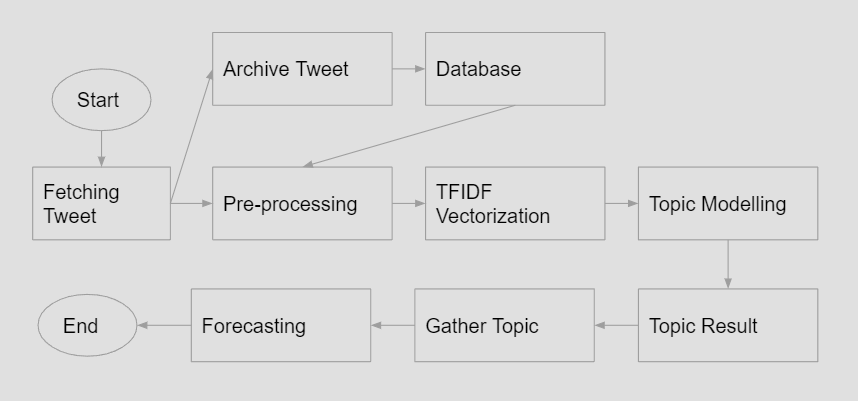
**Model**

Model yang digunakan adalah TF-IDF *vectorizer* dan LDA untuk pemodelan topiknya. Mengingat kami harus memprediksi tren apa yang terjadi di masa depan dirasa lebih tepat ketika menggunakan pendekatan unsupervised dengan tipe *clustering* menggunakan LDA.

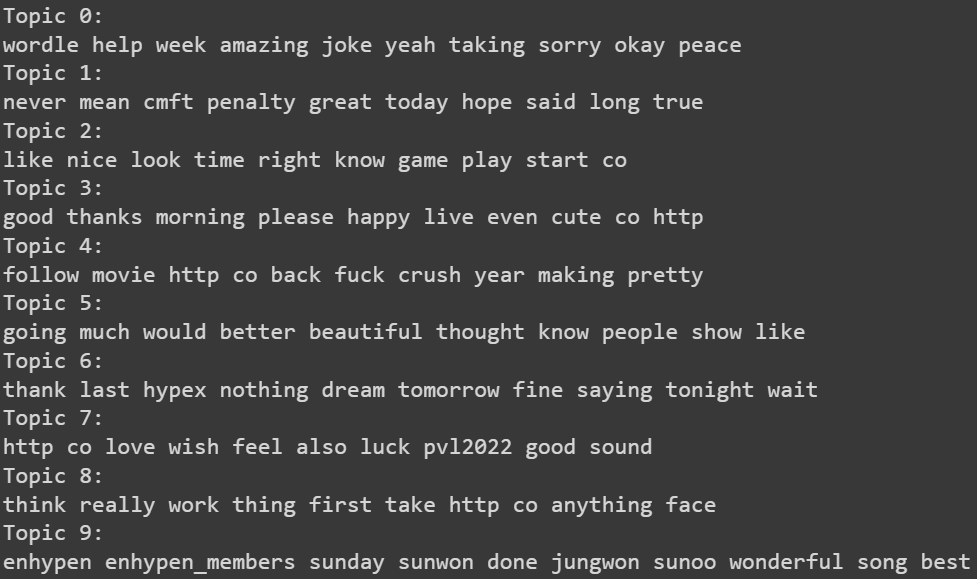
**Workflow**

Workflow yang dilakukan masih bersifat manual

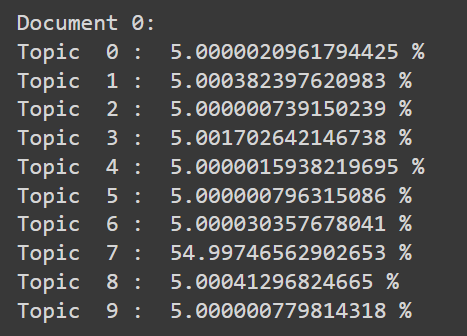
Berikut adalah workflow nya:



Kami melakukan pengambilan data tweet setiap hari dan memodelkan topiknya. Dan menghasilkan output sebagai berikut



Berikut adalah langkah yang perlu dilakukan sebelum melakukan forecasting



Setiap topik memiliki persentase masing-masing. Untuk topik dengan persentase tertinggi akan dicari beberapa kata dengan frekuensi terbanyak. Misalkan di topik 7 kata-kata love, feel, dan good menjadi 3 kata dengan frekuensi terbanyak. 3 kata tersebut akan masuk ke sebuah dataset baru dengan fitur sebagai berikut.

| Date | Word | Frequency | Rank |
| --- | --- | --- | --- |
|  |  |  |  |

Rank adalah hasil feature-engineering. Setelah dataset diatas sudah memiliki record yang cukup panjang barulah dimungkinkan untuk dilakukan *forecasting*. Untuk pengembangan lebih lanjut dari proyek ini, dapat dilakukan pembuatan dashboard menggunakan *python* dan otomasi pada pengambilan data. Workflow bersifat sama untuk 3 jenis data (*Tweet*, Data internal DailySocialId , *web-scraping*).

Kendala dalam melakukan forecast:

*Forecasting* membutuhkan *historical data* yang cukup banyak. API resmi Twitter tidak mengizinkan untuk mengambil *historical data*.

## **BAB IV HASIL PEKERJAAN**

**Web Scraping**

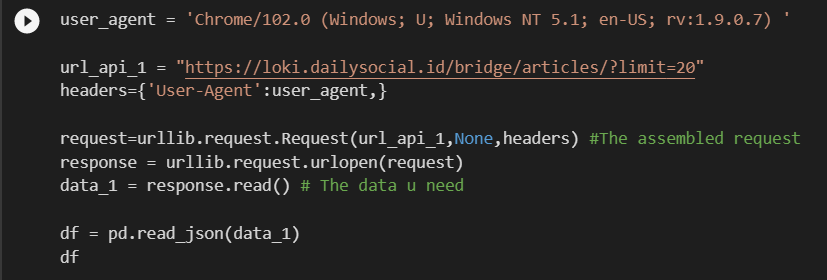
Pengambilan webscrape dari sumber yang ditugaskan tidak bisa dilakukan scrapping secara langsung. Web yang mau diambil data beritanya memproteksi teks sehingga tidak bisa diambil dengan mudah. Karena hal ini web-scraping dilakukan menggunakan hasil pencarian google search. Terdapat beberapa kelemahan menggunakan hasil pencarian dari google yaitu:

* Hasil pencarian tidak bersih(ada web yang tidak dicari dan tanggal yang berbeda dari query search yang digunakan).
* Teks yang terambil hanya headlinenya saja.

**Fetch Data Twitter**

Pengambilan data tweet menggunakan API resmi twitter yang didapatkan dengan mengaktifkan twitter developer account. Kemudian membuat rikues penggunaan API dengan tujuan riset dan edukasi sehingga bisa diapprove dengan segera.

**Akses Data Internal**

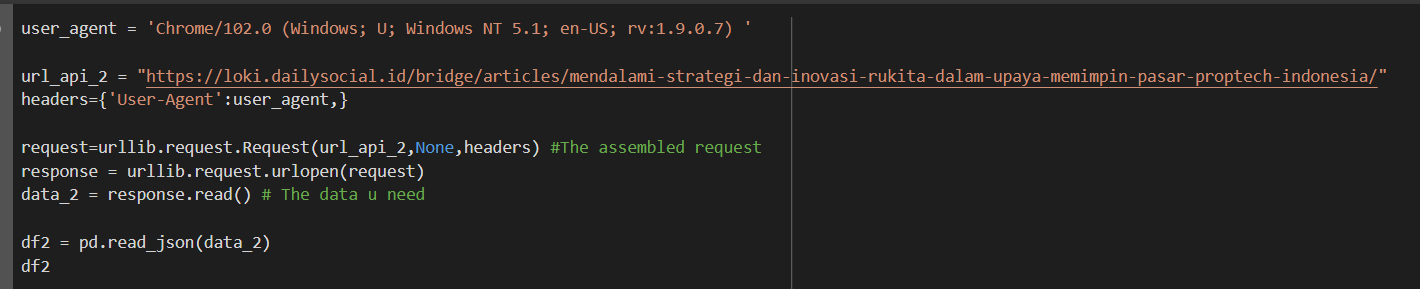
Untuk akses data internal, DailySocial.Id sudah memberikan API nya. Dengan API tersebut kami akan mencoba melakukan akses data internal. Pertama kami melakukan pengecekan apakah API ini dapat diakses dengan cara seperti ini.

Lalu dari kodingan diatas akan menghasilkan output seperti ini



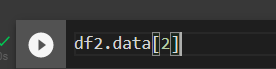
Terbukti dari gambar yang ada diatas, API yang diberikan oleh DailySocial.Id dapat diakses. Jika kami mengecek detail dari data tersebut maka tampilan output nya akan berubah menjadi seperti ini. 

Lalu kami melakukan percobaan untuk melakukan akses untuk artikel yang lebih detil. Dalam contoh kasus ini kami akan mengakses lebih detil terkait artikel dengan judul "Mendalami strategi dan inovasi rukita dalam upaya memimpin pasar proptech indonesia" seperti gambar dibawah ini.

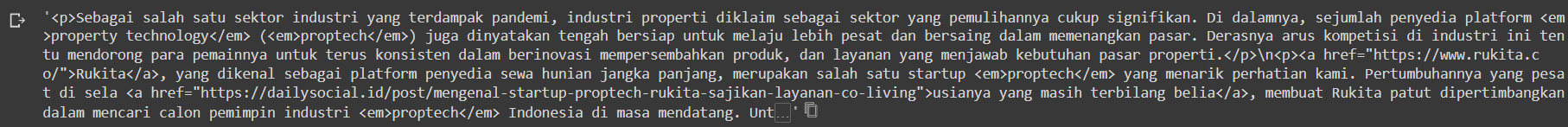


Setelah melakukan kodingan diatas maka tampilan output nya akan menjadi seperti ini 

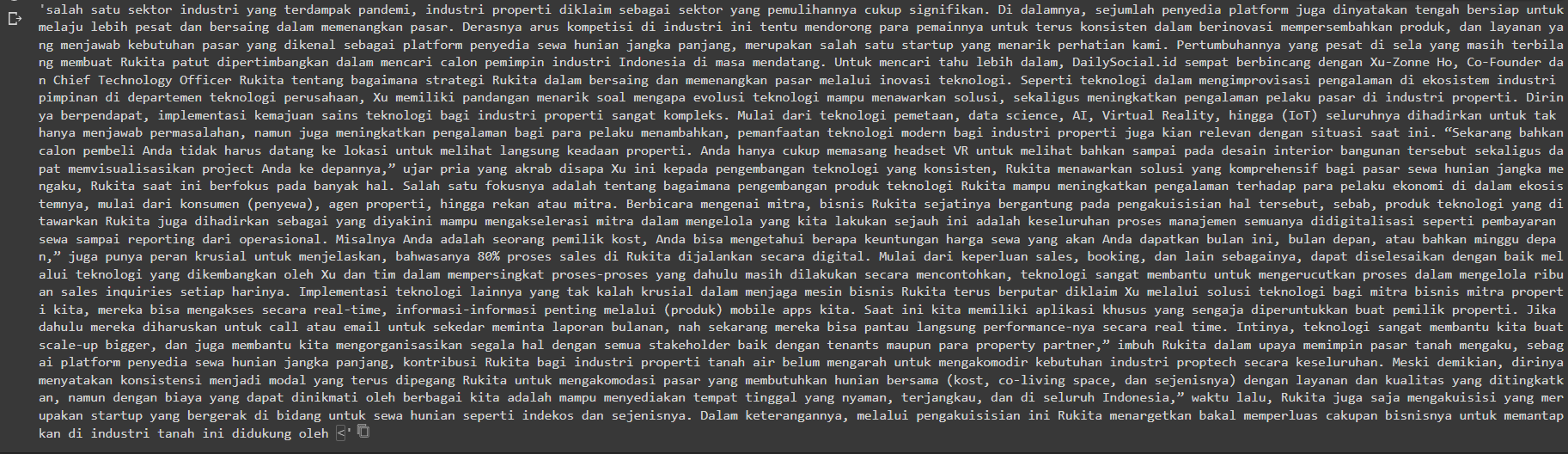
Disini kami akan coba mengambil content yang terdapat di dalam artikel ini dengan cara seperti gambar dibawah ini.



Dari hasil kodingan diatas maka output yang kami terima akan menjadi seperti gambar dibawah ini.

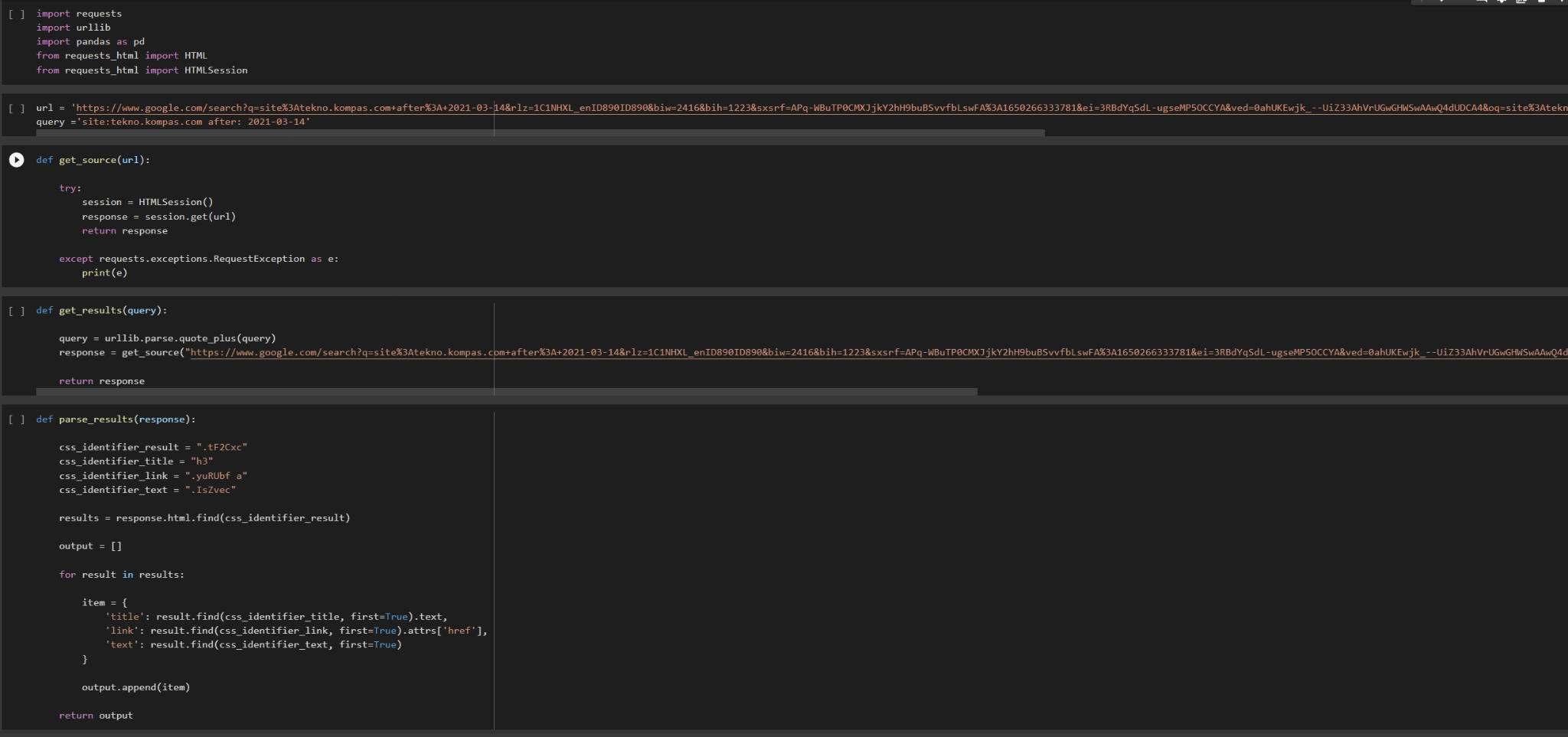
Terlihat dari output yang dihasilkan data masih lah sangat kotor. Perlu dilakukan pembersihan kata supaya data dapat diproses dengan lebih baik. Kita dapat melakukan pembersihan teks kalimat diatas dengan cara seperti gambar dibawah ini.



Jika kita melakukan proses seperti gambar diatas, maka teks yang tadi nya kotor akan berhasil dibersihkan dan jika kita coba tampilan teks nya akan menjadi seperti gambar dibawah ini. 

## **BAB V UJI COBA SEDERHANA**

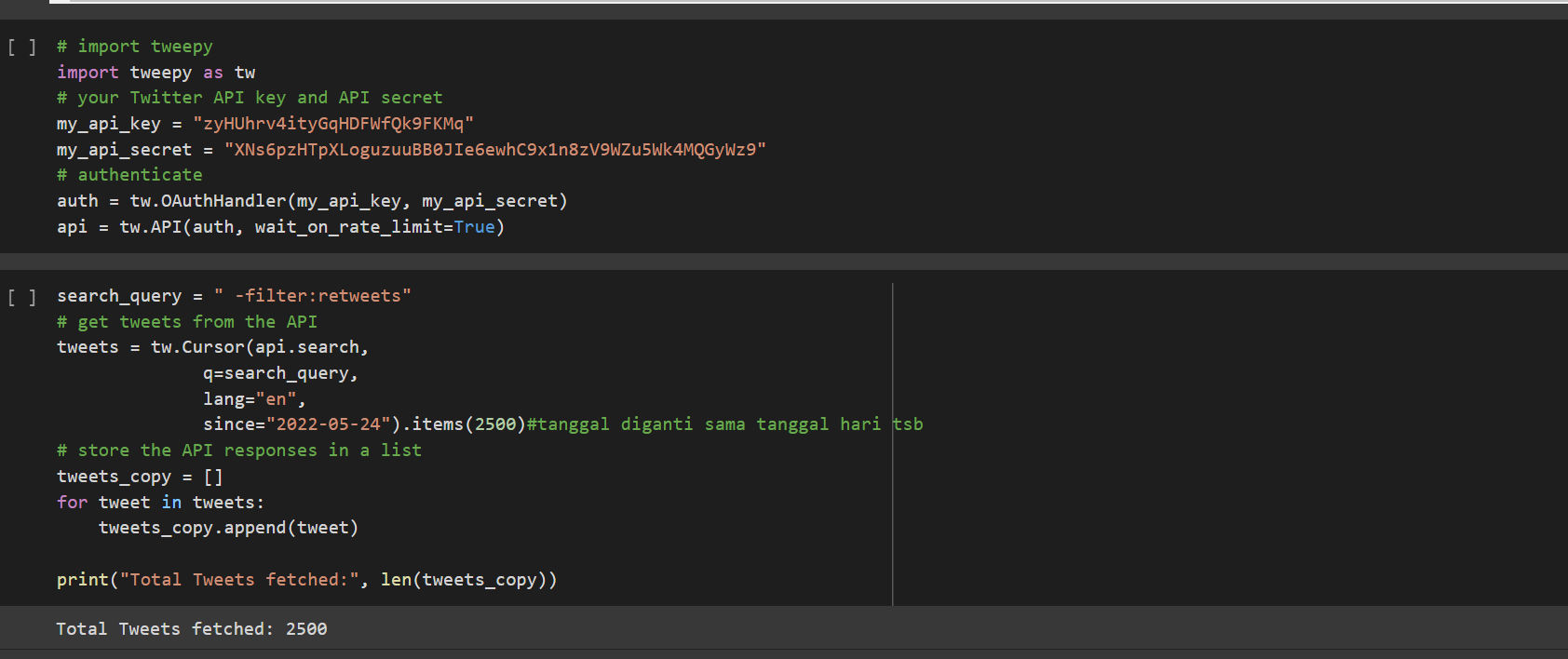
Untuk pengambilan teks hasil google search menggunakan kode sebagai berikut.

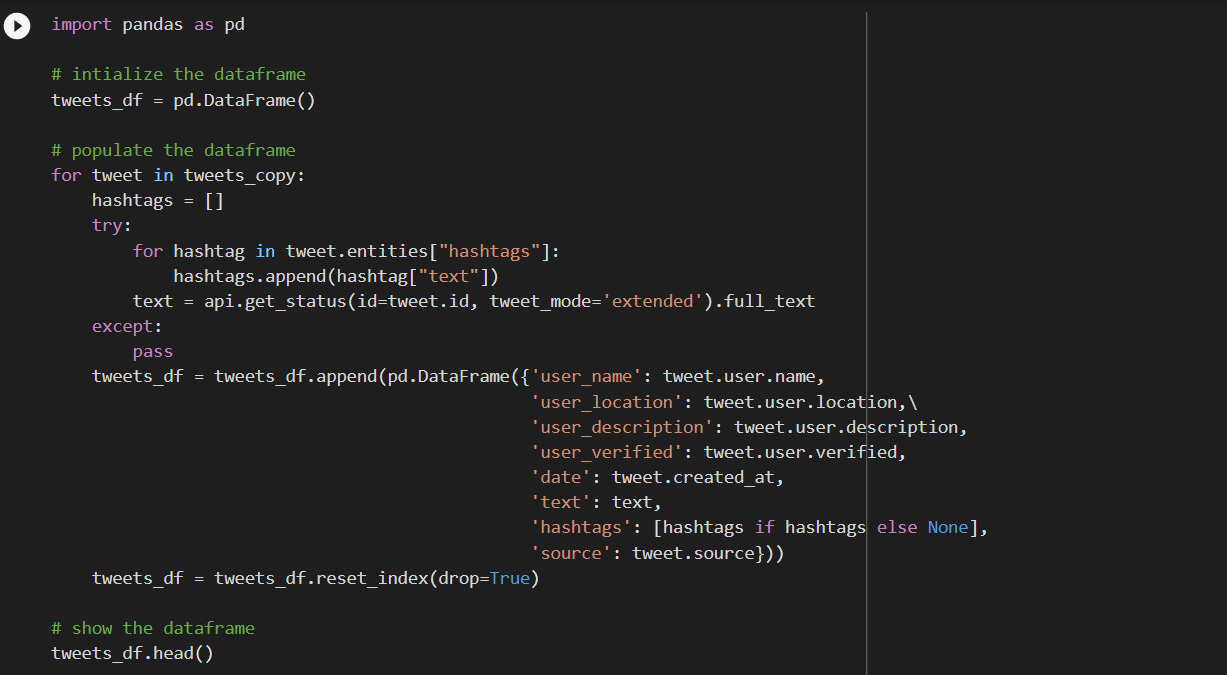


Pengambilan teks dapat diatur dengan variabel query yang digunakan dan website yang dicari.

Hasil scrape dapat dilihat melalui data webscrape yang juga dilampirkan.

Terdapat api key dan api key secret yang perlu digenerate dan dapat dirubah kodenya. Setelah tweet terambil seluruhnya dirubah menjadi dataframe.



****

Kemudian langsung dilakukan preprocessing dan topic modelling sehinga dapat diketahui topik dominan pada hari yang dimaksud.

## **BAB VI KESIMPULAN DAN SARAN**

Untuk data twitter yang diambil tidak memiliki hasil topik yang berarti yang dapat kita lakukan forecast karena kata-kata dominan adalah kata kerja yang tidak mengarah pada topik tertentu. Hal ini dapat hampir pasti disebabkan oleh natur data tweet sendiri yang tidak memliki banyak informasi dan keterbatasan akses untuk mendapatkan lebih banyak data. Untuk hasil webscrape memberikan hasil yang lebih baik namun masih belum bisa memberikan hasil yang sesuai dengan ekspektasi karena hasil model topik masih terlalu general untuk dijadikan sebuah topik.

## **Lampiran: Jadwal Pengerjaan Proyek**

| **Waktu** | **Target** |
| --- | --- |
| Akhir bulan april | Mengolah data internal DailySocialId dan hasil web-scraping. |
| Mei | Memperbesar dataset tweet dan web-scraping |
| Juni minggu ke-3 | Melakukan forecasting dengan sebanyak mungkin data yang telah dikumpulkan. |
|  |  |

## **DAFTAR PUSTAKA**

<https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>

<https://id.wikipedia.org/wiki/Distribusi_Poisson>